H&M *DASHBOARD*

OVERVIEW OF AUTHENTICATION PROCESSES

INTRODUCTION

- This document provides an overview of the authentication process implemented for our system.
- Outlines key steps in the login, refresh token, and logout workflows.
- Intended for developers maintaining or enhancing authentication mechanisms.

PURPOSE AND AUDIENCE

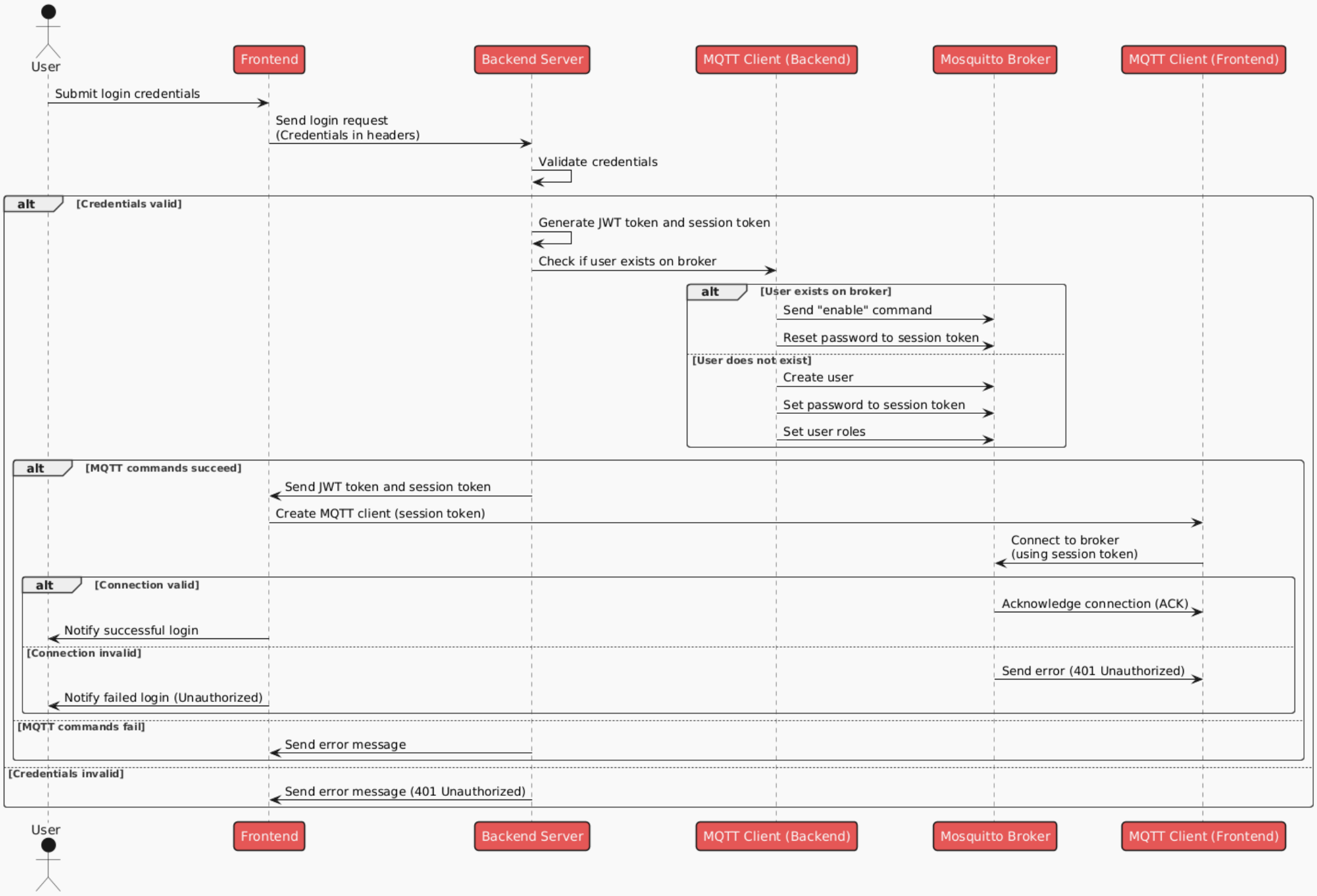
- **Purpose:**
  - Explain the logic and flow behind the authentication process.
  - Detail interactions between components for security and session management.
- **Audience:**
  - Developers working on the **Frontend** and **Backend** systems.
  - Engineers managing the **Mosquitto Broker** and **MQTT Clients**.

AUTHENTICATION WORKFLOW OVERVIEW

- Three primary processes:
  - **Login**: User authentication and session creation.
  - **Token Refresh**: Automatic token renewal to maintain user sessions.
  - **Logout**: Secure user logout and session termination.

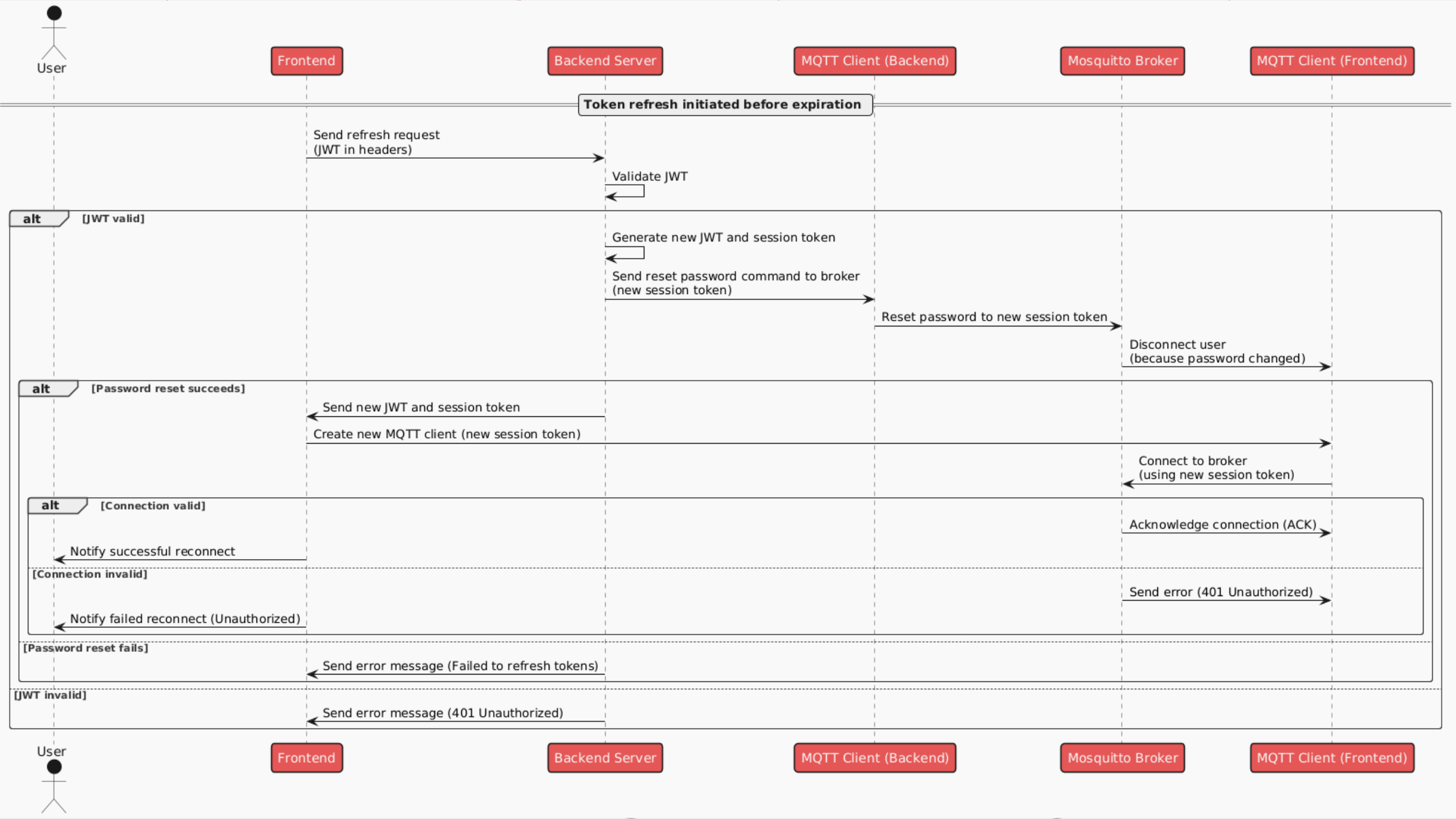
LOGIN PROCESS

- **Workflow Description:**
  - User submits login credentials via the **Frontend**.
  - **Frontend** sends credentials to the **Backend Server**.
  - **Backend** validates credentials and generates **JWT** and **session token**.
  - **MQTT Client (Backend)** interacts with the **Mosquitto Broker**.
  - Returns tokens to the **Frontend** for establishing a new MQTT connection.


User
Frontend
Backend Server
MQTT Client (Backend)
Mosquitto Broker
MQTT Client (Frontend)
Submit login credentials
Send login request
(Credentials in headers)
Validate credentials
alt
[Credentials valid]
Generate JWT token and session token
Check if user exists on broker
alt
[User exists on broker]
Send "enable" command
Reset password to session token
[User does not exist]
Create user
Set password to session token
Set user roles
alt
[MQTT commands succeed]
Send JWT token and session token
Create MQTT client (session token)
Connect to broker
(using session token)
alt
[Connection valid]
Acknowledge connection (ACK)
Notify successful login
[Connection invalid]
Send error (401 Unauthorized)
Notify failed login (Unauthorized)
[MQTT commands fail]
Send error message
[Credentials invalid]
Send error message (401 Unauthorized)
User
Frontend
Backend Server
MQTT Client (Backend)
Mosquitto Broker
MQTT Client (Frontend)

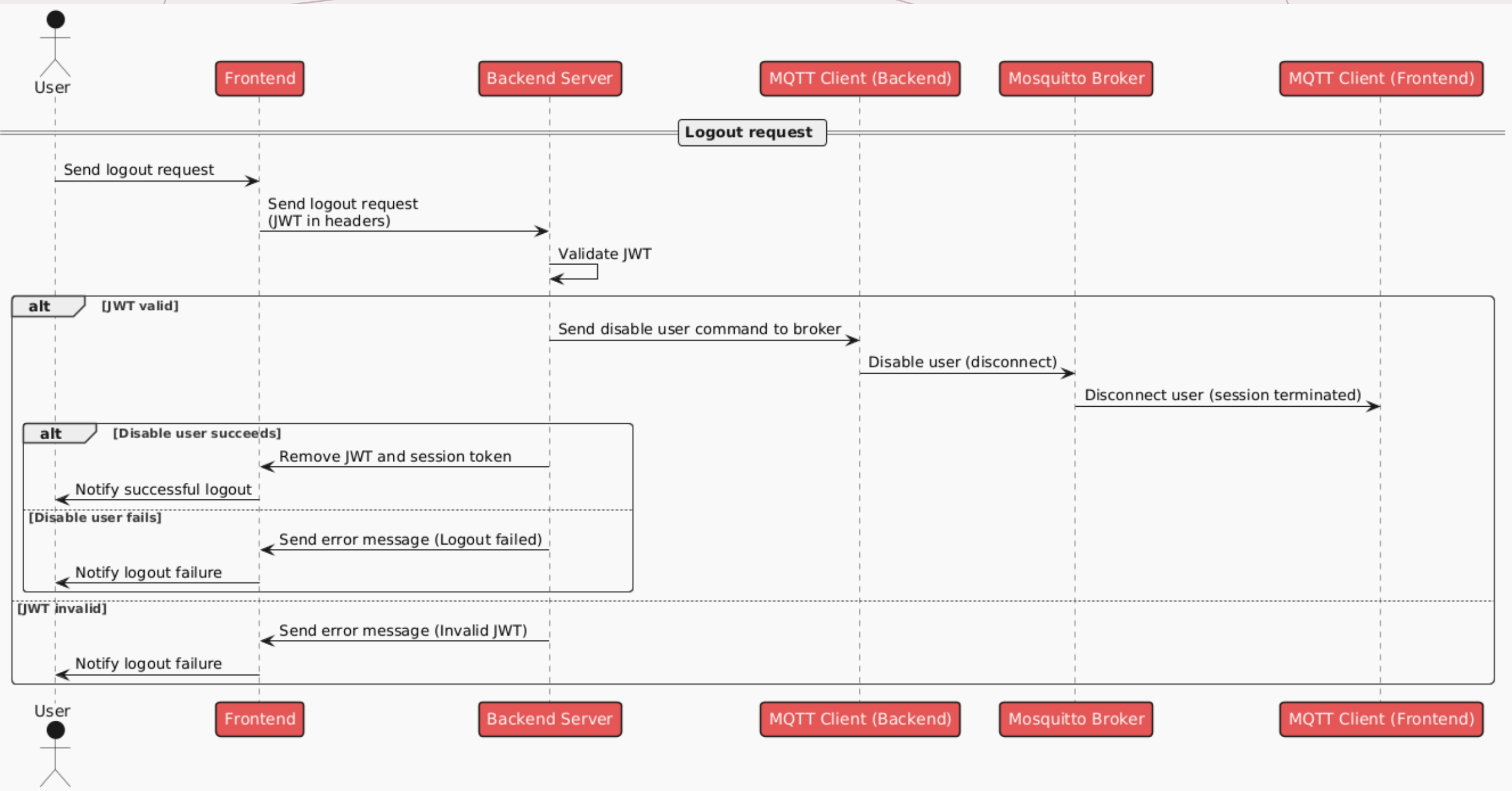
TOKEN REFRESH PROCESS

- **Workflow Description:**
  - Frontend automatically sends refresh request before token expiration.
  - **Backend** validates JWT and generates new tokens.
  - **MQTT Client (Backend)** resets password in **Mosquitto Broker**.
  - User disconnects, and new tokens are returned for a new MQTT connection.


User
Frontend
Backend Server
MQTT Client (Backend)
Mosquitto Broker
MQTT Client (Frontend)
Token refresh initiated before expiration
Send refresh request
(JWT in headers)
Validate JWT
alt
[JWT valid]
Generate new JWT and session token
Send reset password command to broker
(new session token)
Reset password to new session token
Disconnect user
(because password changed)
alt
[Password reset succeeds]
Send new JWT and session token
Create new MQTT client (new session token)
Connect to broker
(using new session token)
alt
[Connection valid]
Acknowledge connection (ACK)
Notify successful reconnect
[Connection invalid]
Send error (401 Unauthorized)
Notify failed reconnect (Unauthorized)
[Password reset fails]
Send error message (Failed to refresh tokens)
[JWT invalid]
Send error message (401 Unauthorized)
User
Frontend
Backend Server
MQTT Client (Backend)
Mosquitto Broker
MQTT Client (Frontend)

LOGOUT PROCESS

- **Workflow Description:**
  - User sends a logout request via the **Frontend**.
  - **Frontend** sends the request to the **Backend**.
  - **Backend** validates JWT and sends disable command to the **Mosquitto Broker**.
  - User is disconnected, and tokens are removed from the **Frontend**.


User
Frontend
Backend Server
MQTT Client (Backend)
Mosquitto Broker
MQTT Client (Frontend)
Logout request
Send logout request
Send logout request
(JWT in headers)
Validate JWT
alt
[JWT valid]
Send disable user command to broker
Disable user (disconnect)
Disconnect user (session terminated)
alt
[Disable user succeeds]
Remove JWT and session token
Notify successful logout
[Disable user fails]
Send error message (Logout failed)
Notify logout failure
[JWT invalid]
Send error message (Invalid JWT)
Notify logout failure
User
Frontend
Backend Server
MQTT Client (Backend)
Mosquitto Broker
MQTT Client (Frontend)